Cluster selection in divisive clustering algorithms*

Sergio M. Savaresi[†], *Daniel L. Boley*[‡], *Sergio Bittanti*[†] *and Giovanna Gazzaniga*[††]

1 Introduction

The problem this paper focuses on is the classical problem of unsupervised clustering of a data-set. In particular, the bisecting divisive clustering approach is here considered. This approach consists in recursively splitting a cluster into two sub-clusters, starting from the main data-set. This is one of the more basic and common problems in fields like pattern analysis, data mining, document retrieval, image segmentation, decision making, etc. ([13], [15]).

Note that by recursively using a bisecting divisive clustering procedure, the data-set can be partitioned into any given number of clusters. Interestingly enough, the so-obtained clusters are structured as a hierarchical binary tree (or a binary taxonomy). This is the reason why the bisecting divisive approach is very attractive in many applications (e.g. in document-retrieval/indexing problems – see e.g. [23]).

Any divisive clustering algorithm can be divided into two sub-problems:

- the problem of selecting which cluster must be split;
- the problem of how splitting the selected cluster.

This paper focuses on the first sub-problem. In particular, in this paper a new method for the selection of the cluster to split is proposed. This method is here presented with

* Paper supported by Consiglio Nazionale delle Ricerche (CNR) short-term-mobility program, and NSF grant IIS-9811229. Thanks are also due to Prof. Gene Golub of Dept. of Computer Science at Stanford.

† Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza L. da Vinci, 32, 20133, Milan, ITALY, {savaresi,bittanti}@elet.polimi.it.

‡ Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CSci, 200 Union St SE, Minneapolis, MN 55455, USA, boley@cs.umn.edu.

†† Istituto di Analisi Numerica - C.N.R, Via Ferrata 1, I-27100 PAVIA, ITALY, gianna@ian.pv.cnr.it.

reference to two specific bisecting divisive clustering algorithms:

- the bisecting K-means algorithm;
- the Principal Direction Divisive Partitioning (PDDP) algorithm.

K-means is the most celebrated and widely used clustering technique (see e.g. [11], [13], [14], [15], [22], [23]); hence it is the best representative of the class of iterative centroid-based divisive algorithms. On the other hand, PDDP is a recently proposed technique ([4], [5], [6], [7]). It is representative of the non-iterative techniques based upon the Singular Value Decomposition (SVD) of a matrix built from the data-set.

The paper is organized as follows: in Section 2 the bisecting K-means and PDDP algorithms are concisely recalled, whereas in Section 3 the method for the selection of the cluster to split is proposed. In Section 4 the problem of evaluating the quality of a set of clusters is considered and some empirical results are presented.

2 Bisecting K-means and PDDP

The clustering approach considered herein is bisecting divisive clustering. Namely, we want to solve the problem of splitting the data-matrix $M = [x_1, x_2, ..., x_N] \in \Re^{p \times N}$ (where each column of M, $x_i \in \Re^p$, is a single data-point) into two sub-matrices (or sub-clusters) $M_L \in \Re^{p \times N_L}$ and $M_R \in \Re^{p \times N_R}$, $N_L + N_R = N$.

This paper focuses on two bisecting divisive partitioning algorithms which belong to different classes of methods: K-means is the most popular iterative centroid-based divisive algorithm; PDDP is the latest development of SVD-based partitioning techniques. The specific algorithms considered herein are now recalled and briefly commented. In such algorithms the definition of "centroid" will be used extensively; specifically, the centroid of M, say w, is given by

$$w = \frac{1}{N} \sum_{j=1}^{N} x_j \ , \tag{1}$$

where x_j is the j-th column of M. Similarly, the centroids of the sub-clusters M_L and M_R, say w_L and w_R, are computed as the average value of their columns.

Bisecting K-means

Step 1. (Initialization). Randomly select a point, say $c_L \in \Re^p$; then compute the centroid w of M, and compute $c_R \in \Re^p$ as $c_R = w - (c_L - w)$.

Step 2. Divide $M = [x_1, x_2, ..., x_N]$ into two sub-clusters M_L and M_R, according to the following rule:

$$\begin{cases} x_i \in M_L & if \quad \|x_i - c_L\| \le \|x_i - c_R\| \\ x_i \in M_R & if \quad \|x_i - c_L\| > \|x_i - c_R\| \end{cases}$$

Step 3. Compute the centroids of M_L and M_R, w_L and w_R.

Step 4. If $w_L = c_L$ and $w_R = c_R$, stop. Otherwise, let $c_L := w_L$, $c_R := w_R$ and go back to Step 2.

The algorithm above presented is the bisecting version of the general K-means algorithm.

This bisecting algorithm has been recently discussed and emphasized in [23] and [25]. It is here worth noting that the algorithm above recalled is the very classical and basic version of K-means (except for a slightly modified initialization step), also known as Forgy's algorithm ([11], [13]). Many variations of this basic version of the algorithm have been proposed, aiming to reduce the computational demand, at the price of (hopefully little) sub-optimality.

PDDP

Step 1. Compute the centroid w of M .

Step 2. Compute the auxiliary matrix $\tilde{M}$ as $\tilde{M} = M - we$, where e is a *N*-dimensional row vector of ones, namely $e = [1,1,1,1,1,..1]$.

Step 3. Compute the Singular Value Decompositions (SVD) of $\tilde{M}$, $\tilde{M} = U\Sigma V^T$, where Σ is a diagonal $p \times N$ matrix, and U and V are orthonormal unitary square matrices having dimension $p \times p$ and $N \times N$, respectively (see [12] for an exhaustive description of SVD).

Step 4. Take the first column vector of U, say $u = U_1$, and divide $M = [x_1, x_2, ..., x_N]$ into two sub-clusters M_L and M_R, according to the following rule:

$$\begin{cases} x_i \in M_L & if \quad u^T(x_i - w) \le 0 \\ x_i \in M_R & if \quad u^T(x_i - w) > 0 \end{cases}.$$

The PDDP algorithm, recently proposed in [5], belongs to the class of SVD-based data-processing algorithms ([2], [3]); among them, the most popular and widely known are the Latent Semantic Indexing algorithm (LSI – see [1], [10]), and the LSI-related Linear Least Square Fit (LLSF) algorithm ([9]). PDDP and LSI mainly differ in the fact that the PDDP splits the matrix with an hyperplane passing through its centroid; LSI through the origin. Another major feature of PDDP is that the SVD of $\tilde{M}$ (Step 3.) can be stopped at the first singular value/vector. This makes PDDP significantly less computationally-demanding than LSI, especially if the data-matrix is sparse and the principal singular vector is calculated by resorting to the Lanczos technique ([12], [17]).

The main difference between K-means and PDDP is that K-means is based upon an iterative procedure which, in general, provides different results for different initializations, whereas PDDP is a "one-shot" algorithm which provides a unique solution, given a data-set. In order to understand better how K-means and PDDP work, in Fig.1a and Fig.1b the partition of a generic matrix of dimension 2×2000 provided by K-means and PDDP, respectively, is displayed. From Fig.1, it is easy to see how K-means and PDDP work:

- the bisecting K-means algorithm splits M with an hyperplane which passes through the centroid w of M, and is perpendicular to the line passing through the centroids w_L and w_R of the sub-clusters M_L and M_R. This is due to the fact that the stopping condition for K-means iterations is that each element of a cluster must be closer to the centroid of that cluster than the centroid of any other cluster.
- PDDP splits M with an hyperplane which passes through the centroid w of M, and is perpendicular to the principal direction of the "unbiased" matrix $\tilde{M}$, which is the translated version of M, having the origin as centroid. The principal direction of $\tilde{M}$ is its direction of maximum variance (see [GV96]).

It is interesting to note that the results of K-means and PDDP are very close, even if the two algorithms differ substantially (a theoretical explanation of this fact is given and

discussed in [21]).
K-means and PDDP algorithms, however, provide a solution only to the first sub-problem of bisecting divisive partitioning: how to split a cluster. The problem of selecting which cluster is the best to be split is left untouched. This will be the topic of the following Section.

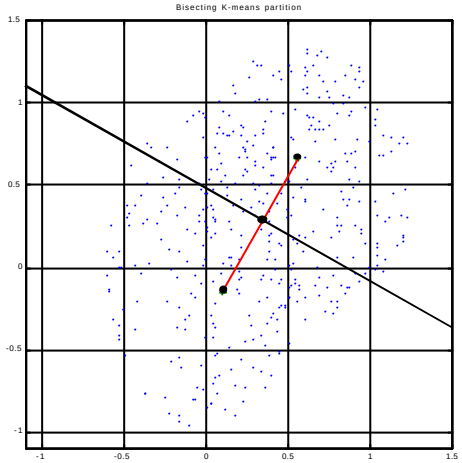

Fig.1a. Partitioning line (bold) of bisecting K-means algorithm. The bullets are the centroids of the data-set and of the two sub-clusters.

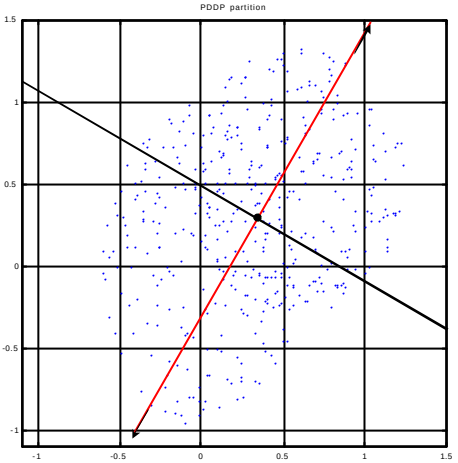

Fig.1b. Partitioning line (bold) of PDDP algorithm. The bullet is the centroid of the data set. The two arrows show the principal direction of $\tilde{M}$.

3 Selecting the cluster to split

The problem of selecting the cluster to split in divisive clustering techniques may have a remarkable impact on the overall clustering results. In the rest of this section a brief overview on the existing approaches will be given in Subsection 3.1; a new method for cluster selection will be presented in Subsection 3.2, and discussed in Subsection 3.3.

3.1. Selecting the cluster to split: a quick overview

The following three classes of approaches are typically used for the selection of the cluster to split ([14] – see also [16]):
(A)complete partition: every cluster is split, so obtaining a complete binary tree;
(B) the cluster having the largest number of elements is split;
(C) the cluster with the highest variance with respect to its centroid

$$\boldsymbol{a}(M) = \frac{1}{N}\sum_{j=1}^{N}\left\|x_j - w\right\|^2 \qquad (2)$$

is split (w is the centroid of data-matrix of the cluster, xj its j-th column, $\|\cdot\|$ is the Euclidean norm).
The above criteria are extremely simple and raw. Criterion (A) is indeed a "non-choice", since every cluster is split: it has the advantage of providing a complete tree, but it

completely ignores the issue of the quality of the clusters. Criterion (B) is also very simple: it does not provide a complete tree, but it has the advantage of yielding a "balanced" tree, namely a tree where the leaves are (approximately) of the same size. Criterion (C) is the most "sophisticated", since it is based upon a simple but meaningful property (the "scatter") of a cluster. This is the reason why (C) is the most commonly used criterion for cluster selection.

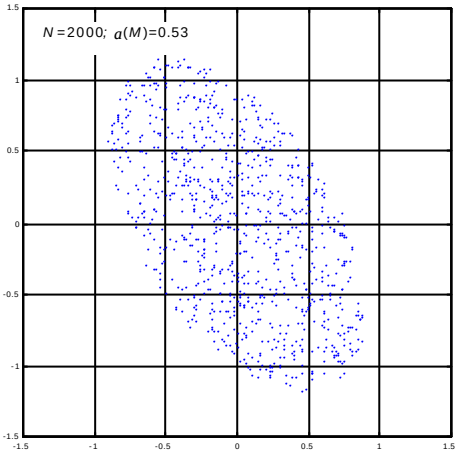

Fig.2a. A data-set with 2000 data-points.

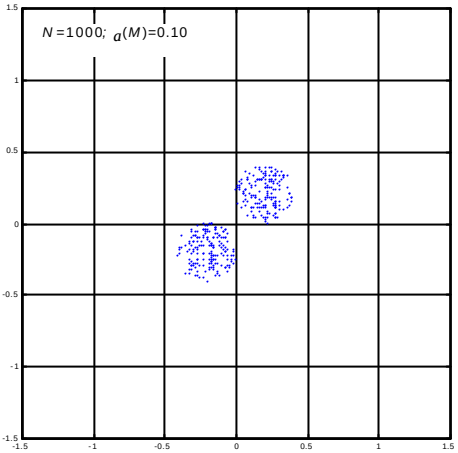

Fig.2b. A data-set with 1000 data-points.

The main limit of the above criteria can be pictorially described with a naive example. In Fig.2 two data-sets are displayed: the first is a matrix of size 2×2000 (Fig.2a); the second is a matrix of size 2×1000 (Fig.2b). By inspecting the two data-sets, it is apparent that the best cluster to split is the second one: it is inherently structured into two sub-clusters. Both criterion (B) and (C), however, would suggest the first one as the best cluster to split: it has the largest number of data-points, and the largest variance ($a(M) = 0.53$ for the first data-set, and $a(M) = 0.10$ for the second data-set).

It is interesting to observe that the main limit of criteria A)-C) is that they completely ignore the "shape" of the cluster, which is known to be a key indicator of the extent to which a cluster is well-suited to be partitioned into two sub-clusters. This simple but crucial observation, however, deserves some additional comments:

- taking into account the "shape" of the cluster is a difficult and slippery task, which inherently requires more computational power than the computation of the simple criterion (2). Henceforth, in computationally-intensive applications the simplicity of criteria A)-C) can be attractive; however, in many applications characterized by a comparative small number of data-points (N) and features (p), a better criterion than A)-C) would be appealing.
- taking into account the "shape" of the cluster requires a wise balancing between an application-specific approach, and a multi-purpose approach. If the criterion is too application-specific it can only be helpful for that application. If too generic (as A)-C) are), it cannot provide high clustering performance.

3.2. Selecting the cluster to split: a new method

Consider a cluster $M = [x_1, x_2, ..., x_N] \in \Re^{p \times N}$, its centroid w, and the vector u ($\|u\| = 1$) which defines the direction along which the cluster should be split, namely:

$$\begin{cases} x_i \in M_L & if \quad u^T (x_i - w) \le 0 \\ x_i \in M_R & if \quad u^T (x_i - w) > 0 \end{cases}.$$

Both in bisecting K-means and PDDP the partition rule is completely described by *w* and *u*. Specifically, in K-means $u = (c_R - c_L) / \|c_R - c_L\|$, and in PDDP, u is the principal eigenvector of $\tilde{M} = M - we$.

The new criterion we propose can be computed as follows:

♦ Project the points of M_L and M_R along the line passing through the centroid, having the direction of *u*:

$$M_L^{\perp} = u^T (M_L - w \cdot e), \; M_R^{\perp} = u^T (M_R - w \cdot e). \tag{3}$$

$M_L^{\perp}$ and $M_R^{\perp}$ are row vectors having the same number of data-points of M_L and M_R, respectively. The elements of $M_L^{\perp}$ are ≤ 0; the elements of $M_R^{\perp}$ are >0.

♦ Normalize $M_L^{\perp}$ and $M_R^{\perp}$, so that they both range from 0 to 1:

$$\tilde{M}_L^{\perp} = M_L^{\perp} / \min(M_L^{\perp}), \; \tilde{M}_R^{\perp} = M_R^{\perp} / \max(M_R^{\perp}).$$

♦ Compute (I_{mL}, I_{cL}) and (I_{mR}, I_{cR}) as:

$$(I_{mL}, I_{cL}) = \left((\tilde{w}_L)^2, \frac{1}{k_L} \sum_{j=1}^{k_L} (\tilde{M}_{Lj}^{\perp} - \tilde{w}_L)^2 \right), \; (I_{mR}, I_{cR}) = \left((\tilde{w}_R)^2, \frac{1}{k_R} \sum_{j=1}^{k_R} (\tilde{M}_{Rj}^{\perp} - \tilde{w}_R)^2 \right),$$

where $\tilde{w}_L$ and $\tilde{w}_R$ are the centroids of $\tilde{M}_L^{\perp}$ and $\tilde{M}_R^{\perp}$, k_L and k_R are the dimensions of $\tilde{M}_L^{\perp}$ and $\tilde{M}_R^{\perp}$, $\tilde{M}_{Lj}^{\perp}$ and $\tilde{M}_{Rj}^{\perp}$ are the *j*-th elements of $\tilde{M}_L^{\perp}$ and $\tilde{M}_R^{\perp}$, respectively.

♦ Compute $(I_m, I_c) = (0.5(I_{mL} + I_{mR}), 0.5(I_{cL} + I_{cR}))$.

♦ Compute the criterion, denoted $g(M)$, as:

$$g(M) = \frac{I_c}{I_m}. \tag{4}$$

If M_a and M_b are two clusters, and $g(M_a) < g(M_b)$, M_a is more suited to be partitioned than M_b.

Note that the meaning of $g(M)$ is intuitive; it is the ratio between the average variance of $\tilde{M}_L^{\perp}$ and $\tilde{M}_R^{\perp}$ around their centroids, and the average squared values of their centroids. $\tilde{M}_L^{\perp}$ and $\tilde{M}_R^{\perp}$ are the normalized sub-clusters M_L and M_R projected along the line passing through the centroid with direction u.

3.3. Discussion

The method for cluster selection presented above can be briefly commented as follows:

- Note that neither the scatter of M (given by (2)) nor the distance between the

centroids of its sub-clusters provide useful information about the shape of the data-set *M*. Indeed, the ratio between scatter and centroid distance is the indicator that really matters. Indeed, if $g(M)$ is small, the cluster is expected to be constituted by two clearly separated sub-clusters, since their scatter is small with respect to their centroids distance. On the other hand, if $g(M)$ is large, the cluster cannot be clearly partitioned, since the two sub-clusters are close and scattered. Criterion $g(M)$ hence is expected to be a concise but good indicator of the shape of the cluster.

- Indicator (4) summarizes the properties of the cluster projected along a 1-dimensional line (defined by u). Of course, this is a limitation with respect to a p-dimensional shape analysis of *M*. However, note that this provides the best compromise between computational complexity and shape-information, since u is the direction of maximum variance of the cluster (this property holds only approximately for K-means - see [21]).

At a first glance, the fact that *M* must be split in order to compute $g(M)$ may appear nonsensical: indeed the role of $g(M)$ is to tell us which cluster must be split. This issue deserves some comments:

- as already said, the calculation of a shape-indicator for *M* is inherently a computational-demanding task. However, among the many different ways of doing this, $g(M)$ has the major advantage that, if *M* is selected, no additional computation are required. Note that this is not guaranteed for a generic shape-indicator (in other words, the efforts spent to compute $g(M)$ can be somehow "recycled").
- if the bisecting clustering recursive procedure is stopped when the data-set has been partitioned into K sub-clusters, it is easy to see that the computation of $g(M)$ has required K-1 "useless" bisecting partitions. However, note that this has little impact on the overall computational balance, since such partitions are made at "leaves-level" (namely they are partitions of small clusters). Low-level partitions are known to be much less demanding than high-level partitions.

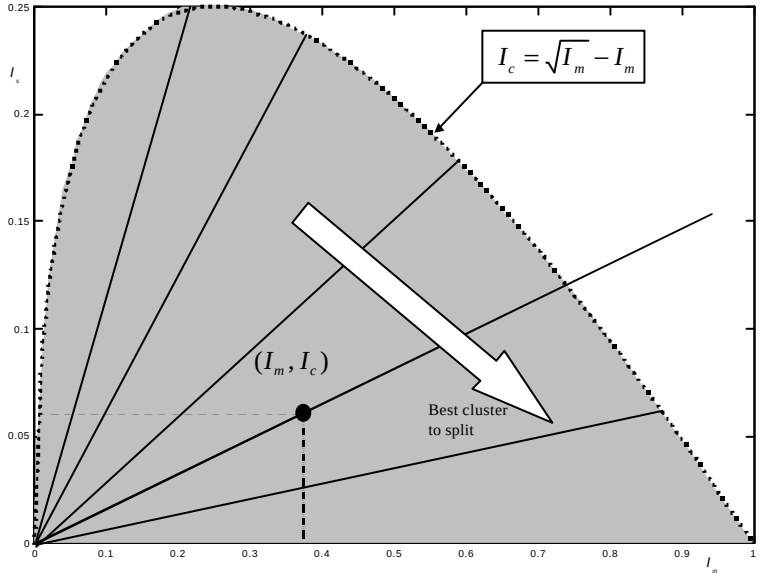

Fig.3. Domain of points (I_m, I_c).

The shape-indicator (4) can be given a simple but interesting graphical interpretation. On the 2-dimensional space where the abscissae is given by the average centroid distance from the splitting line (I_m), and the ordinate is given by the average scatter of the sub-cluster (I_c), $g(M)$ is the slope of the line passing through the origin and the point (I_m, I_c). The smaller this slope is, the more suited M is to be split. Moreover, it can be proven that, whatever M is, the point (I_m, I_c) lies within a compact convex 2-dimensional interval bounded by the lines I_c=0 and $I_c = \sqrt{I_m} - I_m$ (and $0 \le I_m \le 1$). This domain is depicted in Fig.3 (the computation of this is non-trivial; it is extensively described in [20]).

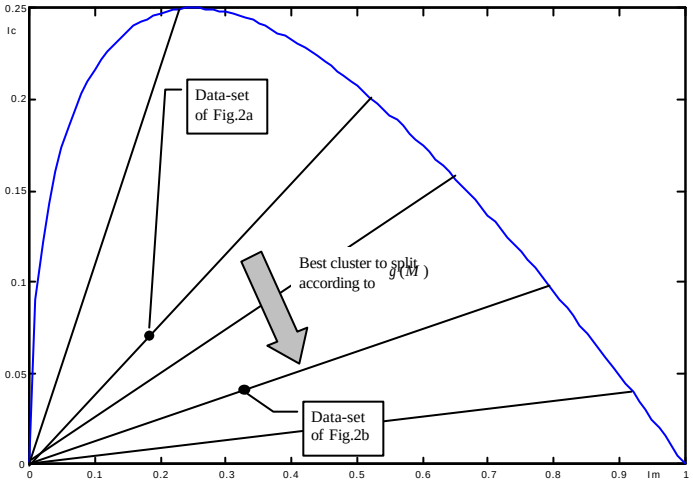

Fig.4. Points (I_m, I_c) computed for the data-sets displayed in Fig.2a and in Fig.2b.

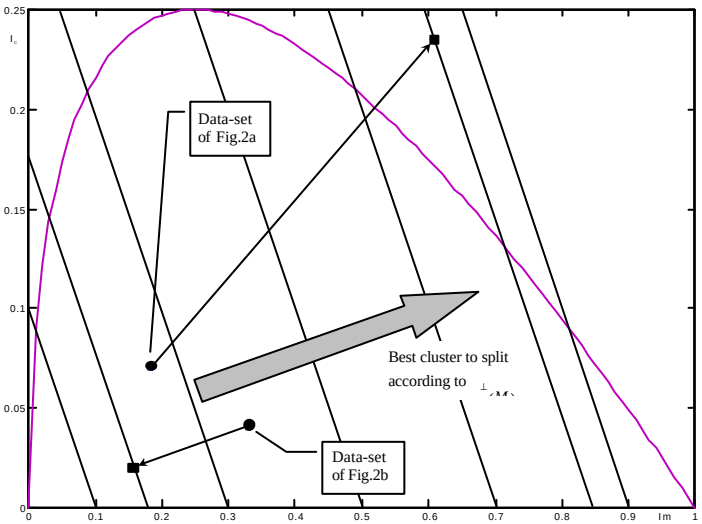

Fig.5. Points (I_m, I_c) (indicated with symbol •) and (2v2 I_m,2v2 I_c) (indicated with symbol ■) computed for the data-sets displayed in Fig.2a and in Fig.2b.

In Fig.4 the points (I_m, I_c) computed for the data-sets displayed in Fig.2a and in Fig.2b are depicted. It is easy to see that the criterion (4) makes the "right" choice (it indicates the data-set in Fig.2b as the most suitable data-set to split).
Criteria $a(M)$ (eq. (2)) and $g(M)$ (eq. (4)) can hardly be compared, since they are substantially different. However, a rough idea of the difference between these two criteria can be obtained as follows. Consider the index $a^{\perp}(M)$, defined as the variance (scatter) of M projected onto a line of direction u, and assume that M is symmetric with respect to the splitting hyperplane (the hyperplane passing through w and perpendicular to u). Henceforth, $v := \min(M_L^{\perp}) = \max(M_R^{\perp})$. Under these assumptions, it is easy to see that the variance of $M^{\perp}$ (the projection of M along u) is equal to the sum of the variance of $M_L^{\perp}$, the variance of $M_R^{\perp}$, and the squared distances of the centroids of $M_L^{\perp}$ and $M_R^{\perp}$ from the centroid of $M^{\perp}$. Given the definitions of I_m and I_c, $a^{\perp}(M)$ can be therefore re-written as: $a^{\perp}(M) = 2v^2(I_m + I_c)$. (5)
Equation (5) is interesting since (even if it holds under some restrictive assumptions) provides a relationship between $g(M)$ and a performance index ($a^{\perp}(M)$) closely related to $a(M)$.
Fig.5 depicts the points (I_m, I_c) (indicated with symbol •) and $(2\sqrt{2}\, I_m, 2\sqrt{2}\, I_c)$ (indicated with symbol ■) computed for the data-sets displayed in Fig.2. It is interesting to see that the criterion (5) sorts the points according to lines having a slope of -45°. Note that they are almost orthogonal to those of criterion (4). According to (5), it is easy to see that the data-set in Fig.2a is the most suitable data-set to split (which is the "wrong" choice, as already remarked at the beginning of this Section).

4 Experimental results

In this section, the selection method proposed in Section 3 will be experimentally tested on a set of real data. This will be done in Subsection 4.2. In Subsection 4.1, the issue of performance evaluation of a clustering process will be preliminary discussed.

4.1. Performance evaluation

When a new clustering algorithm or a modification of an existing algorithm is proposed, a crucial problem is to understand if, and to which extent, this algorithm provides better performance. This is a very subtle and slippery problem.
Any clustering process can be naively described as in Fig.6. The starting point is a set of raw data to be clustered, which are transformed into a matrix of numbers. For the sake of simplicity, we restrict our discussion to the problem in which data samples can be represented by column vectors of numerical attributes which can be assembled into a matrix of numbers. The clustering problem becomes the problem of re-order and partition the columns of the matrix. Two different paths can be followed:

- **The clustering is done by a human expert**. Typically, it is assumed that the partition

made by a human expert is the "correct" partition; obviously, the problem is that a human expert can process only a small amount of data, and that the "expert-evaluation" varies from person to person.

- **The partition is done automatically by a clustering process**. An automatic clustering procedure has the advantage of being able to process billions of data-points; its limit is that it can provide results, which do not make sense to a human expert. Usually, the automatic-clustering process is constituted by three sub-steps: the features selection, the pre-processing, and the application of a clustering algorithm.

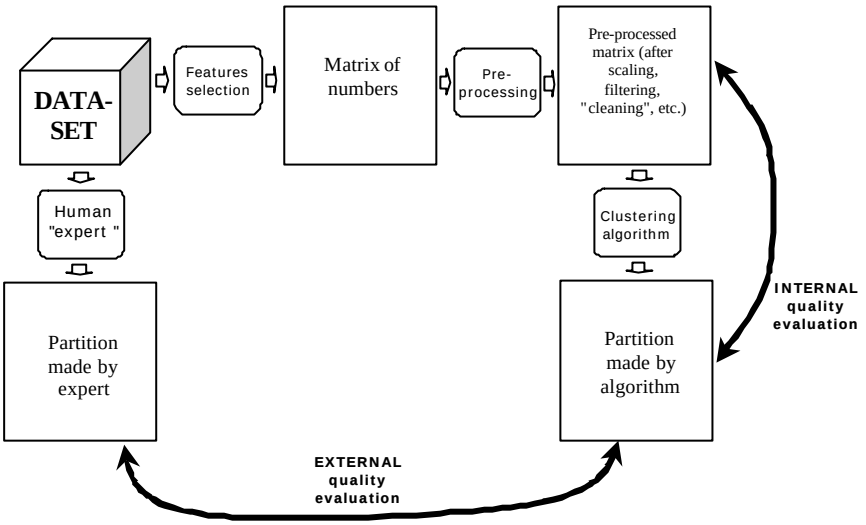

Fig.6. Internal and external quality evaluation.

The evaluation of the results obtained by an automatic clustering procedure can be done in two different ways (see Fig.6):

- **Evaluation of the external quality**. In this case, an automatic clustering procedure is assumed to be good if it provides the same partition yielded by a human expert. A figure of merit for the external quality of the partition hence is a measure of "distance" between the expert-generated and the algorithm-generated partitions. Entropy ([BG+00a]) is a widely used measure of external quality.
- **Evaluation of the internal quality**. In this case, no expert-generated partition is assumed to be available, nor external information. In this case, only the clustering algorithm (not the entire clustering process, including feature selection and pre-processing) is evaluated.

Merits and pitfalls of internal and external figures of merit can be summarized as follows:

- Maximizing the external quality is the final goal in any clustering practical application. As a matter of fact the results obtained by the automatic clustering procedure must be validated by a human expert, in order to be meaningful and actionable. The main limit of external quality evaluation is that it is "subjective", since it is strongly dependent on a human-driven clustering process, and on human-driven steps like features selection and pre-processing. External quality indices hence must be used when dealing with a specific application. External quality instead can be strongly misleading when the goal is a general quality assessment of a clustering algorithm.

- Using internal quality is the best way of measuring the performance of clustering algorithm. Obviously, high internal quality of the algorithm cannot guarantee good results of the overall clustering process in a specific application, since such results also depend on critical steps like features selection and preprocessing.

In this paper we have proposed a general method for improving the performance of bisecting divisive clustering algorithms. This result is general, purely algorithmic, and it is not linked to any specific application. The natural way of evaluating this method hence is to use an internal quality index.

Given the K matrices $\{M_1, M_2, ..., M_K\}$, which constitute a partition of the data-matrix M, the internal quality of the partition can be measured according to the following performance index (see e.g. [15], [22], [23]):

$$J(M_1, M_2, ..., M_K) = \sum_{x_i \in M_1} \|x_i - w_1\|^2 + \sum_{x_i \in M_2} \|x_i - w_2\|^2 + ... + \sum_{x_i \in M_K} \|x_i - w_K\|^2 , \quad (6)$$

where $w_1, w_2, ... w_K$ are the centroids of $\{M_1, M_2, ..., M_K\}$, and x_i is the *i*-th column of M. Note that (6) is a measure of cohesiveness of each cluster to its centroid: the smaller $J(M_1, M_2, ..., M_K)$ is, the better is the partition. This way of measuring the quality of a partition is, however, raw and incomplete. To understand better how an accurate measure of quality should be designed, the general standard definition of a clustering problem is worth to be recalled.

Definition of an unsupervised clustering problem

Given a matrix M, the unsupervised clustering of M into K sub-matrices consists in partitioning M into $\{M_1, M_2, ..., M_K\}$ ($M_i \cap M_j = \varnothing$ if $i \neq j$, $\cup M_j = M$), without a-priori or external information, in order to maximize the similarity among the elements of each sub-matrix (intra-similarity), and to minimize the similarity among elements of different sub-matrices (inter-similarity).

Note that, according to the above definition, the performance index (6) is incomplete: it only evaluates the "intra-similarity", without paying attention to "inter-similarity". Given M and $\{M_1, M_2, ..., M_K\}$, a more sophisticated performance index can be computed as follows.

♦ Compute the scatter, say $\{s_1, s_2, ..., s_K\}$, of each sub-matrix about its centroid. In the scatter the information about "intra-similarity" is condensed. The scatter s_i of M_i is defined as:

$$s_i = \frac{1}{k_i} \sum_{j=1}^{k_i} \|M_{i,j} - w_i\|^2 , \quad (7)$$

where w_i is the centroid of M_i, k_i is the number of columns of M_i, and $M_{i,j}$ is the j-th column of M_i.

♦ Compute the distance, say $\{d_1, d_2, ..., d_K\}$, of each sub-matrix from the the others. In d_i the information about the "inter-similarity" of M_i with respect to the rest of the partition is condensed. The distance d_i can be defined as follows:

$$d_i = \min_j (d_{ij}) \text{, where } d_{ij} = \min_{h,k} (\|M_{i,k} - M_{j,h}\|) \quad \text{j,h,k = 1,2,…,K,} \quad (8)$$

where $\|\cdot\|$ is the Euclidean norm applied to the columns of matrices (note that d_{ij} is the inter-cluster distance used in "single-linkage" agglomeration methods) .

♦ Compute the relative weight, say $\{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_K\}$, of each sub-matrix. It can be defined

as follows:

$$\boldsymbol{d}_i = \frac{k_i}{N}, \tag{9}$$

where k_i is the number of columns of M_i, and $N = \sum_i k_i$.

♦ Compute the performance index $Q(M_1, M_2, ..., M_K)$ as follows:

$$Q(M_1, M_2, ..., M_K) = \sum_{i=1}^{K} \boldsymbol{d}_i \frac{s_i}{d_i}. \tag{10}$$

The smaller $Q(M_1, M_2, ..., M_K)$ is, the better the partition.

Performance index (10) is very intuitive: it is the weighted average (the weights being the relative size of each matrix) of the ratio between scatter and distance. Note that (10) improves (6) since it takes into account the "inter-similarity" among sub-matrices (according to the definition of a clustering problem) and also weights the "importance" of each cluster. In the rest of this section (10) will be used.

We conclude this subsection with a remark. It is worth pointing out that clustering *M* by direct minimization of $Q(M_1, M_2, ..., M_K)$ (or $J(M_1, M_2, ..., M_K)$) would be, conceptually, the best clustering method. Unfortunately, the minimization of $Q(M_1, M_2, ..., M_K)$ requires exhaustive search which is exponential in time with respect to the number of data-points. Note that the clustering algorithms which have been proposed in the literature (including K-means and PDDP) can be interpreted as alternate ways of tackling the problem of minimizing $Q(M_1, M_2, ..., M_K)$. All of them provide a solution with a reasonable computational effort, at the price of some sub-optimality.

4.2. A numerical example

The goal of this subsection is to test the effectiveness of the method for the selection of the cluster to split, presented in Section 3. The method will be tested both on bisecting K-means and PDDP, and the results will be evaluated according to the performance index (10).

Variable	Variable description
1	Major-axis diameter (in arcseconds) from O plate image
2	Integrated magnitude from O plate image
3	Magnitude from O plate image using D-M relation for stars
4	Major-axis position angle (N to E) from O plate image
5	Ellipticity from O plate image
6	Second Moment of O plate image
7	Percent saturation of O plate image
8	Average transmittance of O plate image
9	Mean surface brightness (in mag/asec^2) of O plate image
10	Effective (half-light) radius from O
11	C31 concentration index from O plate.The ratio of the 100% light radius to 50% light radius
12	C32 concentration index from O plate.The ratio of the 100% light radius to 75% light radius
13	C21 concentration index from O plate.The ratio of the 75% light radius to 50% light radius
14	Major-axis diameter (in arcseconds) from E plate image

15	Integrated magnitude from E plate image
16	Magnitude from E plate image using D-M relation for stars
17	Major-axis position angle (N to E) from E plate image
18	Ellipticity from E plate image
19	Second Moment of E plate image
20	Percent saturation of E plate image
21	Average transmittance of E plate image
22	Mean surface brightness (in mag/asec^2) of E plate image
23	Effective (half-light) radius from E
24	C31 concentration index from E plate.The ratio of the 100% light radius to 50% light radius
25	C32 concentration index from E plate.The ratio of the 100% light radius to 75% light radius
26	C21 concentration index from E plate.The ratio of the 75% light radius to 50% light radius
27	E(B-V) determined by bilinear interpolation of Burstein & Heiles [1982] extinction estimates.
28	E(B-V) determined from Schlegel, etal [1998] extinction estimates.
29	O and E imgpars flags (10*Oflag + Eflag).
30	O-E color of the object computed using intergrated magnitudes.
31	O-E color of the object computed using D-M relation magnitudes.
32	Estimated local surface density of MAPS-NGP galaxies (in galaxies/degree^2)

Table 1. Features description.

The data-set we have used as a benchmark is a 32×16000 matrix, built from 16000 objects extracted from a database of the University of Minnesota, consisting in a MAPS-NGP catalog of galaxies images on POSS I (Palomar Observatory Sky Survey) plates within 30 degrees of the North Galactic Pole. The list of the 32 features condensing the information embedded in each image is listed in Table 1. Each feature has been normalized within the range [-1;+1]. Since our goal here is internal-quality evaluation, no further details on the data-set will be given. Detailed information on the data can be found in [8], [19], or at the URL http://lua.stcloudstate.edu/~juan/.

Using the above 32×16000 matrix, three clustering experiments have been done, both for K-means and PDDP. The three experiments only differ on the method used for the selection of the cluster to split, namely:

Method (B): the cluster characterized by the largest number of elements is split (Subsection 3.1);

Method (C): the cluster characterized by the largest scatter is split (Subsection 3.1);

Method (D): the cluster characterized by the lowest value of γ (see (4)) is split, within the set of the 10 clusters having the largest number of elements (Subsection 3.2).

The clustering procedure has been applied iteratively, and stopped when the number of 256 sub-clusters has been reached. After each step, the quality of the partition has been evaluated using (10). The results are displayed in Fig. 7 (partition made using bisecting K-means) and in Fig.8 (partition made using PDDP).

By inspecting the results displayed in Figs.7-8, the following can be said:

- Both for K-means and PDDP splitting algorithms, the method (D) for cluster selection outperforms the traditional methods (B) and (C). The worst performance is, in both cases, achieved by method (B).
- PDDP seems to take more advantages by method (D) than K-means. Probably this is due to the fact that, on this particular set of data, K-means provides better performance than PDDP. The possible improvements hence are more limited.

Even if these results refer to a specific set of data, they are expected to be quite general, since an internal quality of index has been used. Internal indices are known to be much less application-sensitive than external indices. The cluster selection based upon $g(M)$ hence seems to be an effective method to improve the performance of bisecting divisive clustering algorithms.

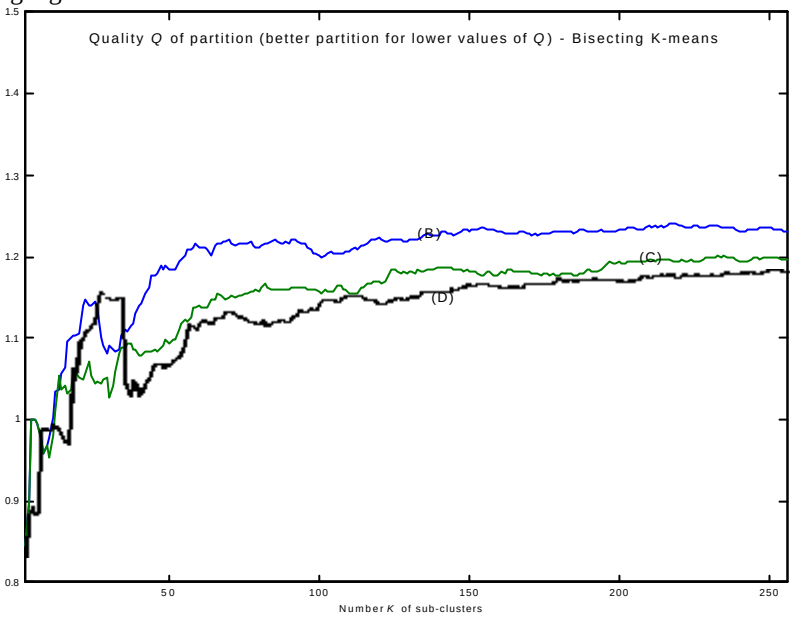

Fig.7. Internal quality evaluation of a partition obtained using bisecting K-means and selection methods (a)-(c).

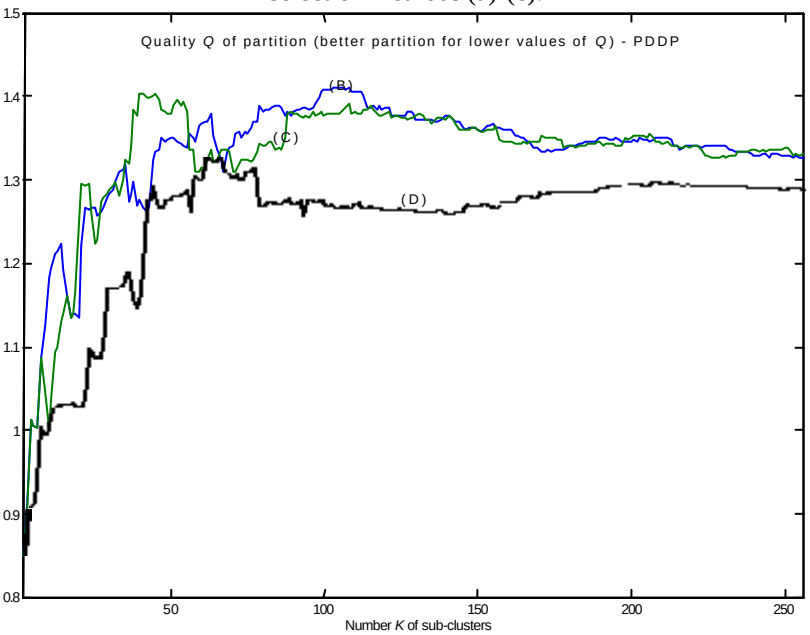

Fig.8. Internal quality evaluation of a partition obtained using PDDP and selection methods (a)-(c).

5 Conclusions

In this paper the problem of clustering a data-set is considered, using the bisecting divisive partitioning approach. This approach can be naturally divided into two sub-problems: the problem of choosing which cluster must be divided, and the problem of splitting the selected cluster. The focus here is on the first problem. A new simple technique for the selection of the cluster to split has been proposed, which is based upon the shape of the cluster. This result is presented with reference to two specific splitting algorithms: the celebrated bisecting K-means algorithm, and the recently proposed Principal Direction Divisive Partitioning (PDDP) algorithm. The problem of evaluating the clustering performance has been discussed, and a test on a set of real data has been done.

References

[1] Anderson, T. (1954). “On estimation of parameters in latent structure analysis”. Psychometrica, vol.19, pp.1-10.

[2] Berry M.W., S.T. Dumais, G.W. O'Brien (1995). “Using Linear Algebra for intelligent information retrieval”. SIAM Review, vol.37, pp.573-595.

[3] Berry, M.W., Z. Drmac, E.R. Jessup (1999). “Matrices, Vector spaces, and Information Retrieval”. SIAM Review, vol.41, pp.335-362.

[4] Boley, D.L. (1997). “Principal Direction Divisive Partitioning”. Technical Report TR-97-056, Dept. of Computer Science, University of Minnesota, Minneapolis.

[5] Boley, D.L. (1998). “Principal Direction Divisive Partitioning”. Data Mining and Knowledge Discovery, vol.2, n.4, pp. 325-344.

[6] Boley, D.L., M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moore (2000). “Partitioning-Based Clustering for Web Document Categorization”. Decision Support Systems, Vol.27, n.3, pp.329-341.

[7] Boley, D.L., M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moore (2000). “Document Categorization and Query Generation on the World Wide Web Using WebACE”. AI Review, Vol.13, n.5-6, pp.365-391.

[8] Cabanela J.E. (1999). Galaxy Properties from a Diameter-limited Catalog. Ph.D. Thesis, University of Minnesota, MN.

[9] Chute, C., Y. Yang (1995). “An overview of statistical methods for the classification and retrieval of patient events”. Meth. Inform. Med., vol.34, pp.104-110.

[10] Deerwester, S., S. Dumais, G. Furnas, R. Harshman (1990). “Indexing by latent semantic analysis”. J. Amer. Soc. Inform. Sci, vol.41, pp.41-50.

[11] Forgy, E. (1965). “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification”. Biometrics, pp.768-780.

[12] Golub, G.H, C.F. van Loan (1996). Matrix Computations (3rd edition). The Johns Hopkins University Press.

[13] Gose, E., R. Johnsonbaugh, S. Jost (1996). Pattern Recognition & Image Analysis. Prentice-Hall.

[14] Jain, A.K., R.C. Dubes (1988). Algorithms for clustering data. Prentice-Hall advance reference series. Prentice-Hall, Upper Saddle River, NJ.

[15] Jain, A.K, M.N. Murty, P.J. Flynn (1999). “Data Clustering: a Review”. ACM Computing Surveys, Vol.31, n.3, pp.264-323.

[16] Karypis, G., E.-H. Han, V. Kumar (1999). "CHAMELEON: A hierarchical

clustering algorithm using dynamic modeling". IEEE Computer, Vol.32, pp.68-75.
[17] Lanczos, C. (1950). “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators”. J. Res. Nat. Bur. Stand, vol.45, pp.255-282.
[18] LaSalle, J.P. (1986). The Stability and Control of Discrete Processes. Springer-Verlag.
[19] Pennington, R.L., R.M. Humphreys, S.C. Odewahn, W. Zumach, P.M. Thurnes (1993). "The Automated Plate Scanner Catalog of The Palomar Sky Survey - Scannig Parameters and Procedures". P.A.S.P, n.105, pp.521-ff.
[20] Savaresi, S.M. (2000). Data Mining: Algorithms and Applications. Laurea Thesis, Università del Sacro Cuore, Brescia (in Italian).
[21] Savaresi S.M., D.L. Boley (2001). “On the performance of bisecting K-means and PDDP”. 1st SIAM Conference on DATA MINING, Chicago, IL, USA, April 5-7, paper n.5, pp.1-14.
[22] Selim, S.Z., M.A. Ismail (1984). “K-means-type algorithms: a generalized convergence theorem and characterization of local optimality”. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.6, n.1, pp.81-86.
[23] Steinbach, M., G. Karypis, V. Kumar (2000). “A comparison of Document Clustering Techniques”. Proceedings of World Text Mining Conference, KDD2000, Boston.
[24] Vidyasagar, M. (1993). Nonlinear Systems Analysis. Prentice-Hall
[25] Wang, J.Z., G. Wiederhold, O. Firschein, S.X. Wei (1997). “Content-based image indexing and searching using Daubechies' wavelets”. Int. J. Digit. Library, vol.1, pp.311-328.